06-28-06

AP ZFW

OIPE MAIL STOP
JUN 26 2006
PATENT & TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/931,296
Filing Date: August 16, 2001
Appellants: DUTTA ET AL.

Primary Examiner:
Macin R. Filipczyk

REPLY BRIEF

"Express Mail" Mailing Label Number EQ 460748030 US

Date of Deposit: June 26, 2006

I hereby certify that this paper or fee is being deposited with the United States Postal Services "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, P. O. Box 1450, Alexandria, Virginia 22313-1450

Darcell Walker, Reg. No. 34,945

This reply brief is in response to the Examiner's Answer filed April 24, 2006 responding to the Appellants' Appeal February 13, 2006.

Response to Examiner's Arguments in Reply Brief

Claims 1-28 are rejected under 35 U.S.C. 102(e) as being anticipated by Bentley et al (US Patent No.: 6,341,291).

Applicants' invention provides a method and system for creating a software storage, retrieval, and execution facility that charts the history and development of software programs. This storage facility would contain collection of legacy software programs that span time from the introduction of computers and software beginning in the 1950's to the explosion of various software products being developed today. The intent is for each software versions to be stored in an executable form.

A user desiring to execute a particular program can access that program via the database and execute the program according to the user's desires. The program accessed by the user may be 20 years old and no longer supported by the creators or suppliers. For example, a user may want to use a FORTRAN program for the 1970's. These programs or specific editions of the program are not readily or commercially available today. With the present invention, the user could retrieve the desired executable edition of the program and perform the desired activity with the program.

In the database, there may be several WINDOWS operating systems, such as WINDOWS 95, WINDOWS 98, WINDOWS 2000 and WINDOWS 2002. All of the programs are WINDOWS operating systems. However, each program is a distinct product. A user may want to use the WINDOWS 95 program. This program is not the current version or the most preferred. But a user may want to use this version. In the present invention, there is not executable connection between this program and any other program in the database.

The Examiner summarizes in the answer that Bentley describes a system used in product design and development to update the project information. To enable the product design and development system, Bentley provides a repository (database) storing a plurality of engineering models. Engineering models include components. Each

component has a unique identifier, a set of fields, each data field has a data type and a data value, a program that interprets and modifies the fields and optionally, a list of dependent components and all components are stored in a database.

After this summary, the Examiner asserts that the terms components, programs and software have the same meaning and are interchangeable. The Examiner further asserts that a database of programs and a database of components is essentially the same. This assumption by the Examiner is inaccurate. First, as the stated in Bentley and described by the Examiner, an engineering model contains a component. Within the components are several attributes. One of the attributes is a program that interprets and modifies the fields.

Referring to Figure 1 of Applicants' invention, item shows various software programs stored in a database. As described in the invention, these software programs represent various programs created over time that were once the state of the art, but now are still capable of executing. Bentley stores engineering models in a database. These engineering models contain a program that enables the user to view and modify the model, and to record any changes in the design such that the design history of the engineering model can be traced. The programs in Bentley are embedded in the model. Based on the objectives described in Bentley and the descriptions in Bentley, it is inaccurate to describe the database in Bentley as a database of programs. It is a database of engineering models, which use a program to facilitate the interpreting, modifying and recording of changes in a particular engineering. Further, for various engineering models, the embedded program in each model could be the same program.

Many of the cites, in Bentley, relied on by the examiner do not support the examiner's assertions. 1) The examiner rejects claims 2 and 12 stating that Bentley discloses multiple software programs, each software program capable of executing in a computing environment (col. 15, lines 29-32). That actual cites reads as follows: "The ProjectBank Server Program 1 also creates, opens, and operates on and maintains the Project History file 3, which contains a list of all changes to components 10 in the project." 2) Regarding claims 3 and 13, the examiner asserts Bentley discloses software programs are stored in software directories according to the type of software program.

(col. 17, lines 63-66). The actual cites reads as follows: "Each history chapter 40 also contains a list 42 of "key components" to designate high-level components contained in the chapter 40. The purpose of the key component list 42 is to increase performance of history chapter 40 searches." 3) Regarding claims 4 and 15, Bentley discloses software directories include directories for operating system programs, application programs and utility programs (col. 18, lines 8-15). The actual cites reads as follows: "ProjectBank Client programs 4 may also save extensive change-description information with each history chapter 40 to document the engineering purpose of the change, or other information as may be relevant. This process does not require special support from the ProjectBank History file 3 since such change-description information can simply be held by a new component 10 that is created for every history chapter 40." Applicants submit that the above described assertions for which the examiner based rejections of the claims are not supported by the cites listed by the examiner. Applicants further submit that the cites are consistent with the objective of Bentley, but not of Applicants present invention.

Applicant submits that some of the features of Applicants' present invention are also features in other computer-implemented applications. However, Applicants submit that the creation of a legacy software database containing executable software programs in a somewhat library format is a novel concept. The ability to retrieve a software program that may be 20 to 25 years of and execute that program is has not been implemented. With the focus on future software developments, Applicants submit that Bentley does not anticipate this concept.

The Examiner's answer appears to broadly interpret the features of Applicants' present invention. Applicants' invention is not merely programs that execute in a computing environment, but legacy software programs. The database of computer programs is not n just a general database of software programs, but legacy software programs as described in the specification. Applicants are only describing these features in the context of the present invention. Applicants are aware that software databases exist. However, Applicants do assert that these databases do not exist in the context as described in the present invention.

Further, the Examiner's answer appears to focus on the system claims and not the method claims. The method claims 10-27 contain steps that are not described or mentioned in Bentley. These steps are illustrated in Figure 7 of the Applicants' application. In particular, with reference to claim 10, the steps of:

submitting a software program request to a database controller;

retrieving the software program identified in the software program request;

determining whether to execute the retrieved software program; and

executing the retrieved software program when the determination is to execute the retrieved software program.

are not described in Bentley.

In view of the above, Applicants respectfully submit that U.S. Patent 6,341,291 to Bentley does not anticipate Applicants' described invention. Contrary to the Examiner's statements that all elements of Applicants' claims are disclosed in the cited reference, the step of submitting a software program request to a database controller;

retrieving the software program identified in the software program request;

determining whether to execute the retrieved software program; and

executing the retrieved software program when the determination is to execute the retrieved software program.

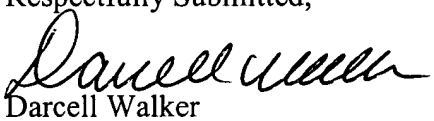
Therefore the 35 U.S.C. § 102(b) rejection of the claims should be withdrawn.

7. CONCLUSION

Applicants submit that all of the pending claims are in condition for allowance. Applicants further submit that the amendments as discussed with the Examiner were for the purpose of further defining the impersonator programs of the present invention. Applicants believe that no additional search should be required in view of the type of amendments Applicants made to the claims. Therefore, withdrawal of the rejections and passage to issuance is respectfully requested.

In view of the above arguments, it is respectfully urged that the rejection of the claims should not be sustained.

Respectfully Submitted,



Darcell Walker
Reg. No. 34,945
9301 Southwest Freeway, Suite 250
Houston, Texas 77074
713-772-1255
June 26, 2006

APPENDIX A

1. A system for maintaining, accessing and executing legacy computer software programs stored in a central location comprising:

a database for storing software programs, developed over a period of time, each software program being stored as a separate and independent software program;

a computing device connected to said database capable of interacting with said database for the purpose of retrieving and executing software programs stored in said database;

a interface device connected to computing device for interacting with said computing device; and

a computing network for connecting said interface device and said computing device.

2. The system as described in claim 1 further comprising in said database, multiple of software programs, each software program being capable of executing in a computing environment.

3. The system as described in claim 2 wherein said software programs are stored in software directories according to the type of software program.

4. The system as described in claim 3 wherein said software directories include directories for operating system programs, application programs and utility programs.

5. (Canceled)

6. The system as described in claim 1 wherein said interface device is a computer terminal.

7. The system as described in claim 1 wherein said computing device is a server.

8. The system as described in claim 7 wherein said server device comprises:

a central processing unit;

a memory;

emulator and simulator programs capable of creating computing environments; and

software retrieval programs for accessing and retrieving software programs stored in said database.

9. The system as described in claim 1 further comprising multiple interface devices capable of interacting with a server simultaneously.

10. A method for maintaining, accessing and executing legacy computer software programs stored in a central location comprising:

creating a database containing multiple types of software programs;

accessing a database that stores multiple types of software programs, developed over a period of time each program capable of executing in a computing environment;

submitting a software program request to a database controller;

retrieving the software program identified in the software program request;

determining whether to execute the retrieved software program; and

executing the retrieved software program when the determination is to execute the retrieved software program.

11. The method as described in claim 10 further comprising after said retrieving step, the step of analyzing the different attributes of the retrieved software program.

12. (Canceled)

13. The method as described in claim 10 wherein said database creating step comprises:

creating a directory for each software program type:

placing each software program in the appropriate directory; and

assigning an identifier to each program.

14. The method as described in claim 13 further comprising the steps of:

creating specific collections of software programs by linking selected software programs together, said collection of programs becoming a software set.

15. The method as described in claim 10 further comprising after said accessing step, the steps of:

retrieving a software programs index, said index containing directories of each type software in the database and a list of each software program in each directory; and

identifying a specific software program in a software directory.

16. A method for accessing and executing software computer programs stored in a central database location, some of said software programs being versions developed over a period of time, said method comprising the steps of:

accessing the database storing multiple types of software programs;

retrieving from a controller connected to the database a software programs options list and a index of software program types and actual software programs;

submitting a request to the software controller containing a selected software programs option and an identified software program;

retrieving from the database an identified software program; and

implementing the selected software option on the identified and retrieved software program.

17. The method as described in claim 16 wherein said implementation step comprises determining whether said selected program option is possible to implement and sending a decline message to the location to the submission when a selected option is not possible to implement on the identified and retrieved software program.

18. The method as described in claim 16 wherein said implementation step comprises creating a computing environment for executing the identified and selected software program.

19. The method as described in claim 18 wherein said computer environment creating step comprises determining the computer hardware and software necessary to implement the selected option on the retrieved software program.

20. The method as described in claim 19 wherein said implementation step occurs in the database controller.

21. The method as described in claim 19 wherein said implementation step occurs in a computing environment located at the software program request submission location.

22. A computer program product in a computer readable medium for accessing and executing software computer programs stored in a central database location comprising:

instructions for accessing the database storing multiple types of software programs;

instructions for retrieving from a controller connected to the database a software programs options list and a index of software types and software program;

instructions for submitting a request to the software controller containing a selected software programs option and an identified software program;

instructions for retrieving from the database an identified software program; and

instructions for implementing the selected software option on the identified and retrieved software program.

23. The computer program product as described in claim 22 wherein said implementation instruction further comprises:

instructions for determining whether said selected program option is possible to implement and;

instructions for sending a decline message to the location to the submission when a selected option is not possible to implement on the identified and retrieved software program.

24. The computer program product as described in claim 22 wherein said implementation instruction further comprises instructions for creating a computing environment for executing the identified and selected software program.

25. The computer program product as described in claim 24 wherein said computer environment creating instruction further comprises instructions for determining the computer hardware and software necessary to implement the selected option on the retrieved software program.

26. The computer program product as described in claim 25 wherein said implementation instructions occur in the database controller.

27. The computer program product as described in claim 25 wherein said implementation instructions occur in a computing environment located at the software program request submission location.

28. A system for maintaining, accessing and executing computer software programs and including a mechanism for storing computer software programs, mechanisms for accessing and executing the stored computer software programs and mechanisms for transmitting and receiving messages over a computer network, said system further comprising:

a computer connectable to a distributed computing environment;

a database of stored computer software programs, said programs comprising a collection of programs spanning a chronological range from the 1950's to the present, each said program capable of being executed in said computer;

a central processor;

a set of operating system programs to enable the execution of programs stored in said database; and

a set of emulation and simulation programs for use in the execution of programs stored in said database.